

Datenbanken II B: DBMS-Implementierung — 10. Übungsblatt: Heap-Dateien, ROWIDs —

Hausaufgaben

Geben Sie die Aufgaben dieses Abschnitts bis Mittwoch, 19.01.2022, 18⁰⁰, über die Übungsplattform in StudIP ab. Schreiben Sie die Lösungen in eine `.txt`-Datei bzw. `.sql`-Datei (kein PDF und schon gar kein Word).

Sie haben in den Übungen Zugriff auf eine Oracle-Datenbank (Version 18c) über die Adminer-Webschnittstelle bekommen:

```
https://dbs.informatik.uni-halle.de/db2b/adminer?  
oracle=oracle-18.4-xe-db2b%2FXEPDB1&username=&db=USERS
```

Sie können diese Aufgaben als normaler Nutzer ausführen. Falls Sie z.B. genauere Informationen zu den belegten Extents wünschen, wäre es aber günstig, Ihren DBA-Account zu verwenden.

- a) **(3 Punkte)** Führen Sie die Datei

```
[http://www.informatik.uni-halle.de/~brass/dbi21/homework/heap.sql]
```

aus (über die Import-Funktion des Adminers). Dies legt eine Tabelle

```
HEAP(ID, A, B, C°)
```

an mit 60 Zeilen. Die Spalten **A** und **B** sind jeweils mit 64 Zeichen langen Strings gefüllt, die Spalte **C** enthält einen Nullwert. In Kapitel 10 der Vorlesung wird erläutert, dass die Zeilen jeweils 139 Byte Speicherplatz benötigen (137 Byte für die Zeile und 2 Byte für den Row Directory Eintrag). Wenn der Block Header 90 Byte groß wäre, würden 58 Zeilen in einen Block passen. Prüfen Sie zunächst mit einer SQL-Anfrage, dass die Tabelle tatsächlich 60 Zeilen enthält. Geben Sie diese Anfrage für Teilaufgabe a) ab.

- b) **(3 Punkte)** Prüfen Sie, ob die Zeilen in der Reihenfolge der IDs gespeichert sind, d.h. ob

```
SELECT * FROM HEAP
```

die Zeilen in der ID-Reihenfolge liefert (sie wurden in dieser Reihenfolge eingefügt). Sie können die Prüfung manuell machen, oder (interessanter) indem Sie das Ergebnis mit der `ROWNUM` in eine Hilfstabelle speichern und diese dann auf Unregelmäßigkeiten untersuchen. Beschreiben Sie in einem kurzen Text, was Sie gemacht haben, und was das Ergebnis war.

- c) **(3 Punkte)** Schreiben Sie eine SQL-Anfrage, die für jeden Block, der mindestens eine Zeile der Tabelle `HEAP` enthält, die Block-Nummer und die Anzahl Zeilen ausgibt. Sortieren Sie das Ergebnis nach der Block Nummer. Geben Sie die Anfrage ab zusammen mit einem kurzen Kommentar, was das Ergebnis war.

Wie im Skript erläutert, brauchen Sie das Package `DBMS_ROWID`. Dies ist in der „Database PL/SQL Packages and Types Reference“ erläutert, die Sie im Bereich „Administration“ finden. In diesem Handbuch ist das Paket in Kapitel 140 erläutert:

[https://docs.oracle.com/en/database/oracle/oracle-database/18/arpls/DBMS_ROWID.html]

Die `ROWID` Pseudospalte ist in der „Oracle SQL Reference“ erläutert:

[<https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/ROWID-Pseudocolumn.html>]

Auch lesenswert ist das Kapitel 12 des „Oracle Concepts Manual“ über „Logical Storage Structures“:

[<https://docs.oracle.com/en/database/oracle/oracle-database/18/cncpt/logical-storage-structures.html>]

- d) **(3 Punkte)** Geben Sie folgenden Befehl ein:

```
ANALYZE TABLE HEAP COMPUTE STATISTICS
```

Eventuell ist das schon automatisch passiert. Früher musste man es manuell eingeben. Sobald die Tabelle analysiert ist, können Sie in der entsprechenden Zeile der Tabelle `TABS` z.B. die Spalte `AVG_ROW_LEN` abfragen für die Länge der Zeilen. Welches Ergebnis bekommen Sie? Möglicherweise stimmt es nicht mit dem errechneten Ergebnis überein. Fragen Sie außerdem die Spalte `CHAIN_COUNT` ab. Dies ist die Anzahl der migrierten Zeilen. Sie müsste 0 sein. Die Spalte `BLOCKS` enthält die „High Water Mark“, also die Anzahl Blöcke, die Zeilen enthalten könnten. Die Spalte `EMPTY_BLOCKS` enthält die zwar reservierten, aber noch völlig unbenutzten Blöcke. Die Spalte `AVG_SPACE` enthält die durchschnittliche Anzahl freier Bytes in den benutzten Blöcken. Genaue Informationen finden Sie in der „Database Reference“ im Eintrag von `ALL_TABLES`:

[https://docs.oracle.com/en/database/oracle/oracle-database/18/refrn/ALL_TABLES.html]

- e) **(3 Punkte)** Führen Sie nun ein verlängerndes Update aus:

```
UPDATE HEAP SET C = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ1234';
```

Anschließend lassen Sie die Tabelle nochmals analysieren:

```
ANALYZE TABLE HEAP COMPUTE STATISTICS
```

Finden Sie nun heraus, ob es migrierte Zeilen gibt, und wenn ja, wie viele. Bestimmen Sie auch die neue Zeilenlänge.

- f) **(3 Punkte)** Welcher Wert für PCTFREE hätte sich mit der Formel der Vorlesung aufgrund der alten und der neuen Zeilenlänge ergeben (um migrierte Zeilen zu vermeiden)? Welchen Wert hat die Tabelle?
- g) **(3 Punkte)** In der Vorlesung haben wir uns gefragt, wann eine migrierte Zeile bei einem „Full Table Scan“ gelistet wird: Dort, wo sie ursprünglich gestanden hat, oder am neuen Speicherort. Prüfen Sie nochmals, ob

```
SELECT * FROM HEAP
```

die Zeilen in der Reihenfolge der IDs ausgibt. Das sollte ja die ursprüngliche Speicher-Reihenfolge sein.

Wiederholungsaufgaben

Die „Wiederholungsaufgaben“ brauchen Sie nicht abzugeben. Beschäftigen Sie sich aber bitte auch mit diesen Aufgaben. Notieren Sie sich Fragen, die Sie gerne in der Übung geklärt haben wollen.

- h) Wie würden Sie in einer mündlichen Prüfung auf folgende Fragen antworten?
- Wie kann man in Oracle die Größe des Speicherbereiches festlegen, der für eine neue Relation reserviert wird? Warum ist es überhaupt nützlich, die erwartete Speichergröße einer Tabelle zu berechnen?
 - Was ist eine ROWID oder TID („tuple identifier“)? Was ist der Zweck von ROWIDs/TIDs? Was sind üblicherweise die wesentlichen Komponenten einer ROWID/TID?
 - Die wichtigste Quelle von ROWIDs sind Indexe. Geben Sie eine sehr abstrakte Beschreibung für einen Index. Was wäre ein (stark vereinfachtes) Interface für den Index-Zugriff?
 - Was sind die Funktionen des Moduls „Row Manager“ in einem DBMS? Was sind die beiden grundlegenden Arten, wie man auf Zeilen zugreifen kann?

- Diskutieren Sie die Vor- und Nachteile stabiler ROWIDs (d.h. die ROWID eines Tupels ändert sich nicht für die ganze Lebensdauer des Tupels).
- Wie kann man die ROWID eines Tupels in Oracle abfragen? Wie bekommt man eine lesbare Repräsentation seiner Komponenten?
- Wie kann man eine Zeile mit gegebener ROWID bei Oracle auswählen?
- Erklären Sie, wie man ROWIDs statt eines Fremdschlüssels nutzen kann und welche Vor- und Nachteile das mit sich brächte.
- Beschreiben Sie ein typisches Blockformat für eine Heap-Datei, wenn die Zeilen variabel lang sein können. Wie kann man den Platz einer gelöschten Zeile für eine neue Zeile wiederverwenden, die ein wenig länger ist, wenn der Block insgesamt noch genug Platz hat (aber nicht in einem Stück)?
- Was passiert, wenn ein Update eine Zeile so verlängert, dass sie nicht mehr in den Block passt? Wie kann man anschließend die Zeile über ihre ROWID finden?
- Was passiert, wenn ein bereits verschobene Zeile ein weiteres Mal aktualisiert wird, und anschließend auch in den neuen Block nicht mehr passt? Wie viele Blockzugriffe braucht man, um die Zeile über ihre ROWID zu finden?
- Warum erfordert das ROWID-Konzept eine minimale Speichergröße für die Zeilen (im wesentlichen die Länge einer ROWID)?
- Angenommen, es gibt viele migrierte Zeilen (verschoben aufgrund verlängernder Updates). Für welche Operationen verschlechtert das die Leistung? Was kann man tun, um das Problem zu lösen?
- Wenn Sie vorhersehen können, dass es für eine Tabelle viele verlängernde Updates gibt, was sollten Sie in Oracle tun?
- Wie kann man einen guten Wert für `PCTFREE` berechnen? Welche Annahmen liegen dieser Berechnung zugrunde?
- Was sind die Konsequenzen eines `PCTFREE`-Wertes, der zu klein oder zu groß ist?
- Wann sollte man `PCTFREE` auf 0 setzen? (Beachten Sie, dass dies nicht der Default-Wert ist.)
- Wenn Sie ein eigenes DBMS entwickeln würden, wie würden Sie das Problem verlängernder Updates lösen? Fällt Ihnen eine Alternative zu `PCTFREE` ein, die weniger oder andere Annahmen benötigt?

Für Interessierte

i) System-Spalten in PostgreSQL sind hier beschrieben:

[<https://www.postgresql.org/docs/current/ddl-system-columns.html>]

Die Spalte `ctid` enthält eine Art ROWID. Allerdings ändert sich diese physische Speicheradresse bei PostgreSQL beim `VACUUM FULL` Befehl. Sie ist damit weniger dauerhaft als in Oracle. Vielleicht interessiert Sie noch diese Diskussion über die Extraktion der Block-Nummer aus dem Wert:

[<https://dba.stackexchange.com/questions/65964/how-do-i-decompose-ctid-into-page-and-row-numbers>]

Informationen zum Format eines Datenblocks in PostgreSQL finden Sie hier:

[<https://www.postgresql.org/docs/current/storage-page-layout.html>]

Vielleicht ist auch diese Seite nützlich:

[<https://postgreshelp.com/postgresql-page-layout/>]

Dieses Kapitel aus „The Internals of PostgreSQL“ beschreibt die Speicherung von Relationen beginnend beim Database Cluster:

[<https://www.interdb.jp/pg/pgsql01.html>]